Claims

[c1] 1. A method for implementing a pre-designed state model, said method comprising:

extracting state information from the state model;

processing said extracted state information;

generating a state code and a state table in response to said processed extracted state information;

compiling said state code to generate a runtime code; and

implementing the state model by running said runtime code while referring to said state table.

[c2] 2. A method as in claim 1 wherein extracting state information from the state model comprises determining what events exist in the state model.

[c3] 3. A method as in claim 1 wherein extracting state information from the state model comprises determining what transitions exist between states within the state model.

[c4] 4. A method as in claim 1 further comprising:

generating an events symbols header in response to a header file; and

generating said state code in response to said processed extracted state information and said events symbols header.

[c5] 5. A method as in claim 4 wherein compiling said state code comprises compiling said state code in response to said events symbols header.

[c6] 6. A method as in claim 1 further comprising:

generating a events symbols header in response to an events configuration file; and

generating said state code in response to said processed extracted state information and said events symbols header.

[c7] 7. A method as in claim 1 further comprising annotating the state model with actions and conditions.

[c8]

8. A method for implementing a pre-designed plurality of state models for a

state machine having an event configuration file, said method comprising:

extracting state information from the plurality of state models;

generating an events symbols header from the event configuration file;

processing said extracted state information in response to said events symbols header;

generating a plurality of state codes and a plurality of state tables in response to said processed extracted state information;

compiling said plurality of state codes using said events symbols header to generate a plurality of runtime codes; and

implementing the state model by running said plurality of runtime codes while referring to said plurality of state tables.

[c9] 9.A method as in claim 8 wherein implementing a pre-designed plurality of state models comprises implementing a cooperating set of run-time controllers.

[c10] 10.A method as in claim 8 further comprising:

generating an events symbols header in response to a header file; and

generating said plurality of state codes in response to said processed extracted state information and said events symbols header.

[c11] 11.A state processor for generating a state table and a runtime code for use in implementing of one or more pre-designed state models, said device comprising:

a state model information provider extracting state model information in response to the one or more state models;

a state information separator generating a state code and the state table in response to the one or more state models; and

a compiler compiling said state code and generating the runtime code.

[c12] 12.A device as in claim 11 further comprising:

an event organizer generating an event symbols header in response to a header file; and

said compiler compiling said state code using said event symbols header.

[c13]
13.A device as in claim 12 wherein said event organizer generates an event

symbols header comprising a centralized list of all events for adding or renaming events.

[c14]   14. A device as in claim 12 wherein said event symbols header comprises global and shared event symbol definitions.

[c15]   15. A device as in claim 12 wherein said header file comprises global and shared event symbol definitions.

[c16]   16. A device as in claim 11 further comprising a runtime library.

[c17]   17. A device as in claim 16 wherein said runtime library comprises a generic state machine component for implementing of event handling.

[c18]   18. A device as in claim 16 wherein said runtime library comprises a time and memory efficient interpreter for processing and handling events.

[c19]   19. A device as in claim 16 wherein said runtime library comprises a scripted dynamic events processor for annotating the one or more state models.

[c20]   A device as in claim 11 wherein said state processor generates a plurality of state tables and a plurality of state codes in response to the one or more state models.